



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/799,301	03/11/2004	Jason Robert McGee	RSW920030263US1	7832

36736 7590 03/09/2007  
DUKE W. YEE  
YEE & ASSOCIATES, P.C.  
P.O. BOX 802333  
DALLAS, TX 75380

EXAMINER
----------

VAUTROT, DENNIS L

ART UNIT	PAPER NUMBER
----------	--------------

2167

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	03/09/2007	PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

**Office Action Summary**

Application No.

10/799,301

Applicant(s)

MCGEE ET AL.

Examiner

Dennis L. Vautrot

Art Unit

2167

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 20 December 2006.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-20 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-20 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_

DETAILED ACTION

***Response to Amendment***

1. The applicants' amendment, filed 20 December 2006, has been received, entered into the record and considered.
2. As a result of the amendment, no claims were amended. Claims 1 – 20 are pending in the application.
3. Applicant's Amendment submitted 20 December 2006 does not overcome the 35 USC101 rejection. Examiner hereby maintains the rejection given on the Office Action dated 20 September 2006.

***Claim Rejections - 35 USC § 101***

4. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.
5. Claim 20 is rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The computer readable media, as defined in the specification on page 14, lines 7 - 11 includes transmission type media, such as radio frequency and light wave transmissions. These are not tangibly embodied in a computer-readable medium, and hence non-statutory. There is always some form of physical transformation within a computer because a computer acts on signals and

Art Unit: 2167

transforms them during its operation and changes the state of its components during the execution of a process. Even though such a physical transformation occurs within a computer, such activity is not determinative of whether the process is statutory because such transformation alone does not distinguish a statutory computer process from a nonstatutory computer process. What is determinative is not how the computer performs the process, but what the computer does to achieve a practical application. See *Arrhythmia*, 958 F.2d at 1057, 22 USPQ2d at 1036.

### ***Claim Rejections - 35 USC § 103***

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

7. Claims 1, 4, 7, 9, 12, 15, and 18 – 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Cool et al.** (hereinafter, **Cool**, US 2004/010786) in view of **White** (US 2003/0014447).

8. Regarding claim 1, **Cool** discloses an apparatus for maintaining compatibility between nodes within a distributed systems management environment, comprising:

a primary node [server], said primary node including a primary storage area, said primary storage [local storage] area including at least two primary storage sections  
(See page 2, paragraph [0016] "The server 102 is a computing system that is

configured to make resources available to other computing systems connected to the network...The server includes local storage in the form of a server data store 110...In particular, the data store 110 includes a server application store 115 for storing application code, and a Web service 112 for making upgraded application code available to other computers via the Web serving software.” The two parts of the data store represent two primary storage sections.); and

a plurality [two or more] of secondary nodes [client computer], each secondary node of said plurality of secondary nodes configured with a plurality of configuration settings having a first format, and each secondary node of said plurality of secondary nodes including a secondary storage area[local storage] (See page 2, paragraph [0015] “...two or more computers, such as a server and a client computer are connected over a network...” and see page 2, paragraph [0017] “The client computer 120 is a computing system configured to execute locally-running applications as well as connect to other computers over the network 105. The client computer 120 also includes local storage in the form of a client data store 106.” Because two or more is disclosed here, it is clear that a plurality of secondary nodes could be attached to the primary.),

convey said plurality of configuration settings having said second format from said second primary storage section to each said secondary storage area of said plurality of secondary nodes (See page 4, paragraph [0031] “The files downloaded are stored in a separate location from the existing version of the application.” These files are in the second format as in the claim.); and wherein each said secondary node of said plurality of secondary nodes is operable to:

receive said plurality of configuration settings [new version] having said second format (See page 4, paragraph [0031] "As illustrated in FIG. 3, the new version of the application (i.e., App Ver 1.0.0.1 320) is stored in a folder separate from the existing version of the application."); and

reconfigure in accordance with said plurality of configuration settings having said second format (See page 4, paragraph [0031] "Thus, when the application is launched again, the application starter 107 will execute the new version of the application rather than the old version.")

**Cool** fails to disclose said primary node is operable to: read said plurality of configuration settings having said first format from a first primary storage section of said at least two primary storage sections; transform said plurality of configuration settings having said first format to a plurality of configuration settings having a second format; and write said plurality of configuration settings having said second format to a second primary storage section of said at least two primary storage sections.

However, **White** discloses said primary node is operable to: read said plurality of configuration settings [data document] having said first format from a first primary storage section of said at least two primary storage sections (See page 6, paragraph [0066] "...when an initial request for a customized document [is] received, the document manager reads a data document from database 422..." The configuration settings are stored in the data document in XML format as disclosed in **White**.);

transform said plurality of configuration settings having said first format [raw data document] to a plurality of configuration settings having a second format [customized

Art Unit: 2167

subscription] (See page 6, paragraph [0066] "The transform is applied to the raw data document so as to generate the customized subscription, organization or presentation level document..."); and

write said plurality of configuration settings having said second format to a second primary storage section of said at least two primary storage sections (See White page 6, paragraph [0066] "See page 6, paragraph [0066] "...and the requested document is written to cache 44." The cache is considered a second storage section because it is separate from the section that holds the original configuration files.)

It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Cool** with that of **White** because they deal with version upgrading of data, and by including the document version generation as disclosed in **White** with the automatic propagation as disclosed in **Cool**, the revised version can more efficiently generated. It is for this reason that one of ordinary skill in the art would have been motivated to include said primary node is operable to: read said plurality of configuration settings having said first format from a first primary storage section of said at least two primary storage sections; transform said plurality of configuration settings having said first format to a plurality of configuration settings having a second format; and write said plurality of configuration settings having said second format to a second primary storage section of said at least two primary storage sections.

Art Unit: 2167

9. Regarding claim 4, **Cool** additionally teaches said primary storage area comprises a master repository, and said secondary storage area [109] comprises a node repository (See page 2, paragraph [0017] "In accordance with the invention, the client application store 109 may contain different versions of the same application" The different versions of the applications are defined in the specification as being stored in both the master and secondary repositories – see page 7, lines 1 – 7. Even by calling the secondary storage area a node repository, what is being stored in the secondary storage area are different versions of the application, as in the reference.)

10. Regarding claim 7, **Cool** teaches an apparatus substantially as claimed. **Cool** fails to teach said plurality of configuration settings having said first format comprises at least one 5.x XSL schema document, and said plurality of configuration settings having said second format comprises at least one 6.x XSL schema document. However, **White** teaches said plurality of configuration settings having said first format comprises at least one 5.x XSL schema document, and said plurality of configuration settings having said second format comprises at least one 6.x XSL schema document. (See page 2, paragraph [0018] "Furthermore, customized versions of the document are created by sequentially applying transforms, in the form, for example, of XSL stylesheets, to intermediate versions of the document.") It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Cool** with that of **White** because they deal with version upgrading of data, and by including the XSL documents as disclosed in **White**, the conversion can use a well



Art Unit: 2167

known efficient method for updating XML documents. It is for this reason that one of ordinary skill in the art would have been motivated to include said plurality of configuration settings having said first format comprises at least one 5.x XSL schema document, and said plurality of configuration settings having said second format comprises at least one 6.x XSL schema document.

11. Regarding claim 9, **Cool** additionally discloses said primary node [102] and said plurality of secondary nodes [other computing systems] comprise a plurality of servers. (See page 2, paragraph [0016] "The server 102 is a computer system that is configured to make resources available to other computing systems connected to the network.")

12. Regarding claim 12, **Cool** discloses a method for maintaining compatibility between a primary node and a plurality of secondary nodes within a distributed systems management environment, comprising the steps of:

conveying said plurality of configuration settings having said second format from said second storage area to each secondary node of said plurality of secondary nodes (See page 4, paragraph [0031] "The files downloaded are stored in a separate location from the existing version of the application." These files are in the second format as in the claim.);

at least one of said each said secondary node receiving said plurality of configuration settings [new version] having said second format (See page 4, paragraph

[0031] "As illustrated in FIG. 3, the new version of the application (i.e., App Ver 1.0.0.1 320) is stored in a folder separate from the existing version of the application."); and

said at least one of said each secondary node reconfiguring in accordance with said plurality of configuration settings having said second format (See page 4, paragraph [0031] "Thus, when the application is launched again, the application starter 107 will execute the new version of the application rather than the old version.")

**Cool** fails to disclose reading a plurality of configuration settings having said first format from a first storage area of said primary node; transforming said plurality of configuration settings having said first format to a plurality of configuration settings having a second format; and writing said plurality of configuration settings having said second format to a second storage area of said primary node.

However, **White** discloses reading a plurality of configuration settings [data document] having a first format from a first storage area of said primary node (See page 6, paragraph [0066] "...when an initial request for a customized document [is] received, the document manager reads a data document from database 422..." The configuration settings are stored in the data document in XML format as disclosed in **White**.);

transforming said plurality of configuration settings having said first format [raw data document] to a plurality of configuration settings having a second format [customized subscription] (See page 6, paragraph [0066] "The transform is applied to the raw data document so as to generate the customized subscription, organization or presentation level document..."); and

writing said plurality of configuration settings having said second format to a second storage area of said primary node (See **White** page 6, paragraph [0066] "See page 6, paragraph [0066] "...and the requested document is written to cache 44." The cache is considered a second storage section because it is separate from the section that holds the original configuration files.)

It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Cool** with that of **White** because they deal with version upgrading of data, and by including the document version generation as disclosed in **White** with the automatic propagation as disclosed in **Cool**, the revised version can more efficiently generated. It is for this reason that one of ordinary skill in the art would have been motivated to include reading a plurality of configuration settings [data document] having a first format from a first storage area of said primary node; transforming said plurality of configuration settings having said first format to a plurality of configuration settings having a second format; and writing said plurality of configuration settings having said second format to a second storage area of said primary node.

13. Regarding claim 15, **Cool** additionally discloses said first [110] and second storage areas [109] comprise a master repository. (See page 2, paragraph [0016] "The server 102 includes local storage in the form of a server data store 110." And see page 2, paragraph [0017] "In accordance with the invention, the client application store 109

may contain different versions of the same application.” The server data store and the client application store are considered to be master repositories.)

14. Regarding claim 18, **Cool** teaches a method substantially as claimed. **Cool** fails to teach said plurality of configuration settings having said first format comprises at least one 5.x XSL schema document, and said plurality of configuration settings having said second format comprises at least one 6.x XSL schema document. However, **White** teaches said plurality of configuration settings having said first format comprises at least one 5.x XSL schema document, and said plurality of configuration settings having said second format comprises at least one 6.x XSL schema document. (See page 2, paragraph [0018] “Furthermore, customized versions of the document are created by sequentially applying transforms, in the form, for example, of XSL stylesheets, to intermediate versions of the document.”) It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Cool** with that of **White** because they deal with version upgrading of data, and by including the XSL documents as disclosed in **White**, the conversion can use a well known efficient method for updating XML documents. It is for this reason that one of ordinary skill in the art would have been motivated to include said plurality of configuration settings having said first format comprises at least one 5.x XSL schema document, and said plurality of configuration settings having said second format comprises at least one 6.x XSL schema document.

15. Regarding claim 19, **Cool** additionally discloses said primary node [102] and said plurality of secondary nodes [other computing systems] comprise a plurality of servers. (See page 2, paragraph [0016] "The server 102 is a computer system that is configured to make resources available to other computing systems connected to the network.")

16. Regarding claim 20, **Cool** discloses a computer program product in a computer readable medium for maintaining compatibility between a primary node and a plurality of secondary nodes within a distributed systems management environment, the computer program product comprising:

fourth instructions for conveying said plurality of configuration settings having said second format from said second storage area to each secondary node of said plurality of secondary nodes (See page 4, paragraph [0031] "The files downloaded are stored in a separate location from the existing version of the application." These files are in the second format as in the claim.);

fifth instructions for receiving said plurality of configuration settings [new version] having said second format (See page 4, paragraph [0031] "As illustrated in FIG. 3, the new version of the application (i.e., App Ver 1.0.0.1 320) is stored in a folder separate from the existing version of the application."); and

sixth instructions for reconfiguring in accordance with said plurality of configuration settings having said second format (See page 4, paragraph [0031] "Thus, when the application is launched again, the application starter 107 will execute the new version of the application rather than the old version.")

**Cool** fails to disclose first instructions for reading a plurality of configuration settings having said first format from a first storage area of said primary node; second instructions for transforming said plurality of configuration settings having said first format to a plurality of configuration settings having a second format; and third instructions for writing said plurality of configuration settings having said second format to a second storage area of said primary node.

However, **White** discloses first instructions for reading a plurality of configuration settings [data document] having a first format from a first storage area of said primary node (See page 6, paragraph [0066] "...when an initial request for a customized document [is] received, the document manager reads a data document from database 422..." The configuration settings are stored in the data document in XML format as disclosed in **White**.);

second instructions for transforming said plurality of configuration settings having said first format [raw data document] to a plurality of configuration settings having a second format [customized subscription] (See page 6, paragraph [0066] "The transform is applied to the raw data document so as to generate the customized subscription, organization or presentation level document..."); and

third instructions for writing said plurality of configuration settings having said second format to a second storage area of said primary node (See **White** page 6, paragraph [0066] "See page 6, paragraph [0066] "...and the requested document is written to cache 44." The cache is considered a second storage section because it is separate from the section that holds the original configuration files.)

It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Cool** with that of **White** because they deal with version upgrading of data, and by including the document version generation as disclosed in **White** with the automatic propagation as disclosed in **Cool**, the revised version can more efficiently generated. It is for this reason that one of ordinary skill in the art would have been motivated to include first instructions for reading a plurality of configuration settings [data document] having a first format from a first storage area of said primary node; second instructions for transforming said plurality of configuration settings having said first format to a plurality of configuration settings having a second format; and third instructions for writing said plurality of configuration settings having said second format to a second storage area of said primary node.

17. Claims 2 and 5 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Cool** in view of **White** as applied to claim 1 above, and further in view of **Greene et al.** (hereinafter **Greene**, US 2002/0198734).

18. Regarding claim 2, **Cool** and **White** teach an apparatus substantially as claimed. **Cool** and **White** fail to teach said primary node comprises a master node [master copy], and said plurality of secondary nodes comprises a plurality of slave nodes [client]. However **Greene** teaches said primary node comprises a master node, and said plurality of secondary nodes comprises a plurality of slave nodes. (See page 40, paragraph [0352] "The optimistic concurrency approach in FIG. 25, on the other hand, depicts the client using a read/write copy that must stay in sync with a master copy in

order for updates to be accepted.” In **Greene**, the master node is where the master copy is located and each of the clients represents a slave node.) It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Cool** and **White** with that of **Greene** because **Green** also deals with distributed systems and maintaining compatibility among them, and by including the master and slave concept of **Greene**, the hierarchy of data transfer among systems becomes better defined. It is for this reason that one of ordinary skill in the art would have been motivated to include said primary node comprises a master node [master copy], and said plurality of secondary nodes comprises a plurality of slave nodes [client].

19. Regarding claim 5, **Cool** and **White** teach an apparatus substantially as claimed. **Cool** and **White** do not explicitly teach said convey operation comprises a synch out operation. **Greene**, however, teaches said convey operation comprises a synch out operation [stay in sync]. (See page 40, paragraph [0352] “The optimistic concurrency approach in FIG. 25, on the other hand, depicts the client using a read/write copy that must stay in sync with a master copy in order for updates to be accepted.”) It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Cool** and **White** with that of **Greene** because **Greene** also deals with distributed systems and maintaining compatibility among them, and by including the syncing concept of **Greene**, the nodes can remain compatible or upgraded with the different versions automatically. It is for this reason that one of ordinary skill in



the art would have been motivated to include said convey operation comprises a synch out operation.

20. Claim 3 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Cool** in view of **White** as applied to claim 1 above, and further in view of **Warrington** (US 2002/0093097). **Cool** and **White** teach an apparatus substantially as claimed. **Cool** and **White** fail to teach said distributed systems management environment comprises a WebSphere environment. However **Warrington** teaches said distributed systems management environment comprises a WebSphere environment. (See page 1, paragraph [0003] "For example, web publishing software such as WEBSPPHERE STUDIO™ from IBM Corporation...both produce visual development environments within which web page authors can design web pages using pre-built components.") It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Cool** and **White** with that of **Warrington** because **Warrington** is also a method of upgrading data versions, and by including the Websphere environment as disclosed in **Warrington**, web page files are specifically included in the type of data that can be updated. It is for this reason that one of ordinary skill in the art would have been motivated to include said distributed systems management environment comprises a WebSphere environment.

21. Claims 6 and 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Cool** in view of **White** as applied to claim 1 above, and further in view of **Parikh** (US 2004/0205162).

22. Regarding claim 6, **Cool** and **White** teach an apparatus substantially as claimed.

**Cool** and **White** fail to teach said first format comprises a WebSphere version 5.x format, and said second format comprises a WebSphere 6.x format. However **Parikh** teaches said first format comprises a WebSphere version 5.x format, and said second format comprises a WebSphere 6.x format. (See page 4, paragraph [0025] "...the application server 306 may be implemented with IBM WebSphere Application Server (WAS), such as Version 5.0 application server (WAS). It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Cool** and **White** that of **Parikh** because they are all related to distributed systems management and by including WebSphere as disclosed in **Parikh**, the apparatus can also work with web page deployment. It is for this reason that one of ordinary skill in the art would have been motivated to include said first format comprises a WebSphere version 5.x format, and said second format comprises a WebSphere 6.x format.

23. Regarding claim 11, **Cool** and **White** teach an apparatus substantially as claimed. **Cool** and **White** fail to teach said distributed systems management environment comprises a WebSphere Application Server. However **Parikh** teaches said distributed systems management environment comprises a WebSphere Application Server. (See page 4, paragraph [0025] "...the application server 306 may be implemented with IBM WebSphere Application Server (WAS), such as Version 5.0 application server (WAS). It would have been obvious to one with ordinary skill in the

Art Unit: 2167

art at the time of the invention to combine the teachings of **Cool** and **White** that of **Parikh** because they are all related to distributed systems management and by including WebSphere as disclosed in **Parikh**, the apparatus can also work with web page deployment. It is for this reason that one of ordinary skill in the art would have been motivated to include said distributed systems management environment comprises a WebSphere Application Server.

24. Claim 8 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Cool** in view of **White** as applied to claim 1 above, and further in view of **Teloh et al** (hereinafter **Teloh**, US 2003/0028521). **Cool** and **White** teach an apparatus substantially as claimed. **Cool** and **White** fail to teach said primary node and said plurality of secondary nodes comprise a plurality of data processing units. However **Teloh** teaches said primary node and said plurality of secondary nodes comprise a plurality of data processing units. (See page 2, paragraph [0016] "The parsed configuration information is then transmitted to the applications, which utilize the configuration information to perform data processing.") It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Cool** and **White** with that of **Teloh** because they are all related to the field of distributed systems management and by including data processing units as disclosed in **Teloh**, the apparatus can store the configuration more efficiently. It is for this reason that one of ordinary skill in the art would have been motivated to include said primary node and said plurality of secondary nodes comprise a plurality of data processing units.

25. Claim 10 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Cool** in view of **White** as applied to claim 1 above, and further in view of **Frazer et al** (hereinafter **Frazer**, US 2005/0055595). **Cool** and **White** teach an apparatus substantially as claimed. **Cool** and **White** fail to teach said primary node and said plurality of secondary nodes comprise a cell. However **Frazer** teaches said primary node [24] and said plurality of secondary nodes [28] comprise a cell. (See page 4, paragraph [0036] "Network 20 includes at least one update station, which in this example is a radio base station 24, operable to transmit software updates across a bi-directional communication link...Subscriber stations 28 can be the customer premises equipment in a wireless local loop for voice and data, ...cellular phones, cable modems...capable of communicating through communication link 32." Because no specific definition of cell is provided in the specification, the examiner interprets a cell to represent a subscriber station as used here.) It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Cool** and **White** with that of **Frazer** because all of the references relate to distributed systems, and in particular updating them, and by including the nodes comprising a cell, a more robust apparatus can be created by allowing wireless devices to be upgraded as well. It is for this reason that one of ordinary skill in the art would have been motivated to include said primary node and said plurality of secondary nodes comprise a cell.

26. Claims 13 and 16 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Cool** in view of **White** as applied to claim 12 above, and further in view of **Greene**.

27. Regarding claim 13, **Cool** and **White** teach a method substantially as claimed. **Cool** and **White** fail to teach said primary node comprises a master node [master copy], and said plurality of secondary nodes comprises a plurality of slave nodes [client]. However **Greene** teaches said primary node comprises a master node, and said plurality of secondary nodes comprises a plurality of slave nodes. (See page 40, paragraph [0352] "The optimistic concurrency approach in FIG. 25, on the other hand, depicts the client using a read/write copy that must stay in sync with a master copy in order for updates to be accepted." In **Greene**, the master node is where the master copy is located and each of the clients represents a slave node.) It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Cool** and **White** with that of **Greene** because **Green** also deals with distributed systems and maintaining compatibility among them, and by including the master and slave concept of **Greene**, the hierarchy of data transfer among systems becomes better defined. It is for this reason that one of ordinary skill in the art would have been motivated to include said primary node comprises a master node [master copy], and said plurality of secondary nodes comprises a plurality of slave nodes [client].

28. Regarding claim 16, **Cool** and **White** teach a method substantially as claimed. **Cool** and **White** do not explicitly teach the conveying step comprises a synching out operation. **Greene**, however, teaches the conveying step comprises a synching out

operation [stay in sync]. (See page 40, paragraph [0352] "The optimistic concurrency approach in FIG. 25, on the other hand, depicts the client using a read/write copy that must stay in sync with a master copy in order for updates to be accepted.") It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Cool** and **White** with that of **Greene** because **Greene** also deals with distributed systems and maintaining compatibility among them, and by including the syncing concept of **Greene**, the nodes can remain compatible or upgraded with the different versions automatically. It is for this reason that one of ordinary skill in the art would have been motivated to include the conveying step comprises a synch out operation.

29. Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Cool** in view of **White** as applied to claim 12 above, and further in view of **Warrington** (US 2002/0093097). **Cool** and **White** teach a method substantially as claimed. **Cool** and **White** fail to teach said distributed systems management environment comprises a WebSphere environment. However **Warrington** teaches said distributed systems management environment comprises a WebSphere environment. (See page 1, paragraph [0003] "For example, web publishing software such as WEBSHERE STUDIO™ from IBM Corporation...both produce visual development environments within which web page authors can design web pages using pre-built components.") It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Cool** and **White** with that of **Warrington** because **Warrington** is also a method of upgrading data versions, and by including the

WebSphere environment as disclosed in Warrington, web page files are specifically included in the type of data that can be updated. It is for this reason that one of ordinary skill in the art would have been motivated to include said distributed systems management environment comprises a WebSphere environment.

30. Claim 17 is rejected under 35 U.S.C. 103(a) as being unpatentable over **Cool** in view of **White** as applied to claim 12 above, and further in view of **Parikh**. **Cool** and **White** teach a method substantially as claimed. **Cool** and **White** fail to teach said first format comprises a WebSphere version 5.x format, and said second format comprises a WebSphere 6.x format. However **Parikh** teaches said first format comprises a WebSphere version 5.x format, and said second format comprises a WebSphere 6.x format. (See page 4, paragraph [0025] "...the application server 306 may be implemented with IBM WebSphere Application Server (WAS), such as Version 5.0 application server (WAS). It would have been obvious to one with ordinary skill in the art at the time of the invention to combine the teachings of **Cool** and **White** that of **Parikh** because they are all related to distributed systems management and by including WebSphere as disclosed in **Parikh**, the apparatus can also work with web page deployment. It is for this reason that one of ordinary skill in the art would have been motivated to include said first format comprises a WebSphere version 5.x format, and said second format comprises a WebSphere 6.x format.

***Response to Arguments***

31. Applicant's arguments filed 20 December 2006 have been fully considered but they are not persuasive.

32. In response to applicant's argument that there is no suggestion to combine the references, the examiner recognizes that obviousness can only be established by combining or modifying the teachings of the prior art to produce the claimed invention where there is some teaching, suggestion, or motivation to do so found either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. See *In re Fine*, 837 F.2d 1071, 5 USPQ2d 1596 (Fed. Cir. 1988) and *In re Jones*, 958 F.2d 347, 21 USPQ2d 1941 (Fed. Cir. 1992). In this case, the initial reference, **Cool et al.**, is directed towards upgrading a software application to a different version than what is currently stored on a remote location. In the background information, **Cool et al.** explains why a user might be reluctant to upgrade (see **Cool et al.** page 1, paragraph [0001] and [0002] "Upgrading software applications that have been installed is a much more difficult process than it seems. Software developers encounter multiple problems when releasing an upgrade to a software application. One serious problem is the fear that upgrading one application may adversely affect another installed application in some unpredictable way...If one software application is upgraded and changes the shared library, that change, although unintended, could result in unpredictable behavior by another software application that references the same shared library." While **Cool et al.** goes on to explain a process of upgrading that reduces the



burden to the user, these reasons also provide motivation for not forcing the upgrade, as in the instant application. One with ordinary skill in the art at the time the invention was made would have been looking for a way to keep the compatibility between systems that were not upgraded, recognizing the reasons for not wanting to upgrade, and therefore would have been motivated to apply the teachings of **White et al.** The "customized version" of a document discussed in **White et al.** is representative of a different version of a configuration file, as discussed in the instant application. (See **White et al.** page 1, paragraph [0005] where a disclosure is made, from a co-assigned application of "a system and method for compiling data that defines components to be configured into a personal computer system." This is describing configuration files, and a person with ordinary skill in the art at the time the invention was made would have realized the benefits of translating the data to various versions of the configuration file based on the needs of the requesting node.

Finally, another reason is offered for combining the references in **White et al.**, on page 1, paragraph [0007] "Historically, responding to the demand for personalized versions of data documents has necessitated the development of customized software code to transform a baseline document into the form requested by a client." Here, the baseline document represents the version of the configuration files found on the primary node, with the personalized version being the different format as requested by the secondary node. By combining the transformation system as disclosed in **White et al.** to the node upgrading teachings of **Cool et al.**, the customized version of the configuration files can be created and sent to the other nodes. **White et al.** is making it

possible for the various versions of the software to exist on different nodes by having the configuration files (or raw data) being transformed to the appropriate version needed, as requested.

Based on at least these reasons, examiners feel there was proper motivation to combine **Cool et al.** with **White et al.**

33. In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). Specifically, on page 12, first incomplete paragraph of the arguments dated 20 December 2006 the following argument was made: "White is not related to, does not discuss, and would have no reason to ...convey the plurality of configuration settings having the second format from a second primary storage section of the primary node to a secondary storage area of each of a plurality of secondary nodes." Examiner respectfully disagrees that White would need to show the transferring of the configuration settings, as this rejection is based on a combination of **White et al.** and **Cool et al.** Pointing to the lack of a teaching in one document is not proper when the rejection was not based on one document alone. The other part of the argument, that **White et al.** does not disclose the transformation at a primary node, is shown in **White et al.** as pointed out in the rejection, but to clarify, the "document manager" can be interpreted to be the "primary node".

34. In response to applicant's argument that the examiner's conclusion of obviousness is based upon improper hindsight reasoning, it must be recognized that any judgment on obviousness is in a sense necessarily a reconstruction based upon hindsight reasoning. But so long as it takes into account only knowledge which was within the level of ordinary skill at the time the claimed invention was made, and does not include knowledge gleaned only from the applicant's disclosure, such a reconstruction is proper. See *In re McLaughlin*, 443 F.2d 1392, 170 USPQ 209 (CCPA 1971).

35. The remaining arguments were based on the alleged deficiencies in the 103 rejection for the independent claims, which have already been addressed.

### ***Conclusion***

36. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

Art Unit: 2167


the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Dennis L. Vautrot whose telephone number is 571-272-2184. The examiner can normally be reached on Monday-Friday 9:00-6:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, John Cottingham can be reached on 571-272-7079. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

Dv  
23 February 2007



JOHN COTTINGHAM  
SUPERVISORY PATENT EXAMINER  
TECHNOLOGY CENTER 2100